UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/748,098	12/21/2000	Nicholas J. Kelsey	20880-05093; Case 5093	4360

758 7590 04/13/2005

FENWICK & WEST LLP
SILICON VALLEY CENTER
801 CALIFORNIA STREET
MOUNTAIN VIEW, CA 94041

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 04/13/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/748,098	KELSEY ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 January 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-55* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-55* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 December 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *20 December 2004*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-55 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS as received on 12/20/2004 and Amendment as received on 1/24/2005.

Claim Objections

3. Claim 48 is objected to because of the following informalities: Please replace "an hard" with --a hard--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 46-55 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

6. Claim 46 recites the limitations "the first thread state" and "the second thread state". There is insufficient antecedent basis for these limitations in the claim.

Claim Rejections - 35 USC § 102

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for

patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1-3, 13-17, 19, 21-24, 29-32, and 42-46 are rejected under 35 U.S.C. 102(e) as being anticipated by Joy et al, U.S. Patent Number 6,542,991 (as applied in the previous Office Action and herein referred to as Joy).

8. Referring to claim 1, Joy has taught a computer based system for switching between program contexts comprising:

A processor (Joy figure 3, column 8 lines 14-67) capable of having a first program thread and a second program thread in an execution pipeline having a thread selection hardware (Joy figure 6, column 13 lines 5-23, column 15 lines 4-7);

A first set of data storage devices capable of storing a first thread state of said processor (Joy figure 5 number 510 column 13 lines 5-23);

A second set of data storage devices capable of storing a second thread state of said processor (Joy figure 5 number 512 column 13 lines 5-23); and

A hardware thread scheduler for identifying which of said program threads said processor executes (Joy figure 6 column 15 lines 4-7) and configurable to allocate available processing time of the processor among at least the first and second threads by causing thread-switching according to a predetermined fixed schedule. See Fig.6, column 2, lines 40-45, and column 3, lines 28-31, and note that the scheduler schedules threads based on cache misses. This scheduling is predetermined in that the system was designed, before runtime, to switch based on cache misses. That is, before the system even begins running a program, it is known that the scheduler will schedule based on cache misses because that is how it was designed. This specific

scheduling is also fixed in that if the scheduler is to switch based on cache misses, then if will switch based on cache misses. It will not all of the sudden decide to not switch based on cache misses. Once the hardware is built and programmed to operate as desired, its operation is fixed. It should be noted that the examiner is only referring to the control logic scheduler which performs switching based on cache misses (column 3, lines 28-31).

9. Referring to claim 17 Joy has taught a computer based system for switching between program contexts comprising:

A pipelined processor (Joy figure 3, column 8 lines 14-67) capable of having a first program thread and a second program thread in an execution pipeline having a thread selection hardware (Joy figure 6, column 13 lines 5-23, column 15 lines 4-7);

A first set of data storage devices capable of storing a first thread state of said processor (Joy figure 5 number 510 column 13 lines 5-23);

A second set of data storage devices capable of storing a second thread state of said processor (Joy figure 5 number 512 column 13 lines 5-23); and

A hardware thread scheduler for identifying which of said program threads said processor executes (Joy figure 6 column 15 lines 4-7) and configurable to allocate available processing time of the pipelined processor among at least the first and second states according to a fixed schedule (Joy figure 6, column 2 lines 40-45; the priority of the scheduler would be fixed, thus giving it a fixed schedule, since it would operate the same way each time it encounters the same threads);

Wherein said thread selection hardware in the pipelined processor switches between said first and second thread state between consecutive instruction cycles in response tot the hardware

thread scheduler identifying which of said program threads said processor executes (Joy column 3 lines 45-56, column 6 lines 15-35 column 15 line 52-column 16 line 5, and Fig.1B, 2B, and 2C; after the cycle ends, the second thread will now be in control for the next cycle).

10. Referring to claims 2 and 31 Joy has taught wherein said first state is the state of the processor during the execution of the first program thread (Joy column 3 line 66-column 4 line 35).

11. Referring to claims 3 and 32 Joy has taught wherein said second state is the state of the processor during the execution of the second program thread (Joy column 3 line 66-column 4 line 35).

12. Referring to claims 13 and 42 Joy has taught wherein said processor is capable of restoring said second state of said processor during execution of said first program thread (Joy column 3 lines 45-56, column 6 lines 15-35).

13. Referring to claims 14 and 43 Joy has taught wherein said processor is capable of storing said second state of said processor during execution of said first program thread (Joy column 3 lines 3-10; the information of state 2 will still be stored during the execution of state 1).

14. Referring to claims 15 and 44 Joy has taught wherein said first set of data storage devices comprises registers shared by a plurality of threads (Joy column 3 lines 3-10).

15. Referring to claims 16 and 45 Joy has taught wherein the fixed schedule is one of a fixed strict schedule, a semi-flexible strict schedule, and a loose strict schedule (Joy figure 6, column 2 lines 40-45; the priority of the scheduler would be fixed, thus giving it a fixed schedule, since it would operate the same way each time it encounters the same threads).

16. Referring to claim 19 Joy has taught a computer based method for switching between program contexts in a multithreading pipelined processor (Joy figure 3, column 8 lines 14-67) having a hardware thread selector and an execution pipeline, the method comprising:

Storing a first context of said processor in a first set of data storage devices, the first context corresponding to a first program thread (Joy figure 5 number 510 column 13 lines 5-23);

Storing a second context of said processor in a second set of data storage devices, the second context corresponding to a second program thread (Joy figure 5 number 512 column 13 lines 5-23);

Switching the processor from executing the first program thread to executing the second program thread (Joy figure 6 column 15 lines 4-7) between the end of an execution cycle and before the beginning of a next consecutive execution cycle by coupling the execution pipeline from the first set of data storage devices to the second set of storage devices via the hardware thread selector (Joy column 3 lines 45-56, column 6 lines 15-35 column 15 line 52-column 16 line 5, and Fig.1B, 2B, and 2C; after the cycle ends, the second thread will now be in control for the next cycle).

17. Referring to claim 21 Joy has taught further comprising:

Identifying which of the said program threads said processor executes according to an execution schedule (Joy figure 6 column 15 lines 4-7).

18. Referring to claim 22 Joy has taught further comprising allocating available processing time of the processor among at least the first and second threads according to the execution schedule (Joy figure 6, column 2 lines 40-45).

19. Referring to claim 23 Joy has taught wherein the allocating comprises dividing the available execution time into a plurality of quanta, each quanta corresponding to a number of instruction cycles for execution of a thread (Joy figure 6, column 2 lines 40-45).

20. Referring to claim 24 Joy has taught wherein at least one quanta corresponds to a thread that is scheduled to execute periodically after a fixed number of execution cycles (Joy figure 6, column 2 lines 40-45).

21. Referring to claim 29, Joy has taught wherein said thread selection hardware in the embedded pipelined processor switches between said first and second thread state after the end of the execution of a first program instruction in the first thread and before the beginning of the execution of a second program instruction (Joy column 3 lines 45-56, column 6 lines 15-35; the thread that is idle would have completed a thread, and the second thread would not be able to execute a thread until the switch occurs).

22. Referring to claim 30, Joy has taught wherein said processor is an embedded pipelined processor. See column 7, lines 52-54.

23. Referring to claim 46, Joy has taught a computer based method for switching between program contexts in a multithreading pipelined processor (Joy figure 3, column 8 lines 14-67) having a hardware thread selector and an execution pipeline, the method comprising:

Storing a first context of said processor in a first set of data storage devices, the first thread state corresponding to a first program thread (Joy figure 5 number 510 column 13 lines 5-23);

Storing a second context of said processor in a second set of data storage devices, the second thread state corresponding to a second program thread (Joy figure 5 number 512 column 13 lines 5-23);

Switching the processor from the first thread state to the second thread state by coupling the execution pipeline from the first set of data storage devices to the second set of storage devices via the hardware thread selector (Joy column 3 lines 45-56, column 6 lines 15-35 column 15 line 52-column 16 line 5, and Fig.1B, 2B, and 2C; after the cycle ends, the second thread will now be in control for the next cycle) according to a predetermined fixed execution schedule. See Fig.6, column 2, lines 40-45, and column 3, lines 28-31, and note that the scheduler schedules threads for execution based on cache misses. This scheduling is predetermined in that the system was designed, before runtime, to switch based on cache misses. That is, before the system even begins running a program, it is known that the scheduler will schedule based on cache misses because that is how it was designed. This specific scheduling is also fixed in that if the scheduler is to switch based on cache misses, then if will switch based on cache misses. It will not all of the sudden decide to not switch based on cache misses. Once the hardware is built and programmed to operate as desired, its operation is fixed. It should be noted that the examiner is only referring to the control logic scheduler which performs switching based on cache misses (column 3, lines 28-31).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

24. Claims 4, 20, 33, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joy in view of Borkenhagen et al, U.S. Patent Number 6,567,839 (as applied in the previous Office Action and herein referred to as Borkenhagen).

25. Referring to claims 4, 20, 33, and 47 Joy has not taught wherein said processor switches between said first and second states by changing a state selection register.

Borkenhagen has taught wherein said processor switches between said first and second states by changing a state selection register (Borkenhagen column 16 lines 38-49).

It would have been obvious to one of ordinary skill in the art at the time of the invention to use a register to change the threads of a multithreaded system. The system of Joy must have some mechanism to control the thread switch, but does not disclose details of this logic. Borkenhagen does disclose a particular embodiment for this logic and states reasoning to use this type of switching logic for such reason as assigning certain threads priority, which ties into the Joy system which has hard real time threads (Borkenhagen abstract, column 5 line 66-column 6 line 11). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to look to Borkenhagen to use a register to change the threads of a multithreaded system if they want to do a detailed design since Joy does not describe that logic function.

26. Claims 5-9, 18, 25-28, 34-38, and 48-55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joy in view of Ramakrishnan et al, U.S. Patent Number 6,085,215 (as applied in the previous Office Action and herein referred to as Ramakrishnan).

27. Referring to claims 5, 25-26, 34, 48, and 52-53 Joy has not explicitly taught wherein said hardware thread scheduler includes:

A thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread;

A HRT scheduler for regularly scheduling said HRT thread in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT in a predetermined time. However, Ramakrishnan has taught wherein said thread scheduler includes:

A thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread (Ramakrishnan figures 2, 2a, and 2b, abstract, column 5 line 54-column 6 line 8, column 9 lines 9-21, column 15 line 39-column 16 line 6, column 8 lines 47-53);

A HRT scheduler for regularly scheduling said HRT thread in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT in a predetermined time (Ramakrishnan figures 2, 2a, and 2b, abstract, column 5 line 54-column 6 line 8, column 9 lines 9-21, column 15 line 39-column 16 line 6, column 8 lines 47-53).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have a thread scheduler that handles real time threads so that real time threads will be executed when needed for critical systems needing real time operations. Ramakrishnan has taught this need for scheduling and has taught a solution with real time thread scheduling with general purpose threads and real time threads (Ramakrishnan column 1 lines 10-32, column 3 line 25-column 4 line 6, column 5 line 54-column 6 line 8). This will allow systems requiring real time operations to execute the required functions when needed, and the real time threads will have priority. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the

invention to have a thread scheduler that handles real time threads to allow for real time threads to be completed when required.

28. Referring to claims 6 and 35 the combination of Joy and Ramakrishnan has taught wherein said time quanta is at least one instruction cycle (Ramakrishnan figures 2, 2a, and 2b, abstract, column 5 line 54-column 6 line 8, column 9 lines 9-21, column 15 line 39-column 16 line 6, column 8 lines 47-53; any switching would require some time, so it would take at least one unit of time, or one cycle).

29. Referring to claim 18 the combination of Joy and Ramakrishnan has taught wherein said time quanta is exactly one instruction cycle (Ramakrishnan figures 2, 2a, and 2b, abstract, column 5 line 54-column 6 line 8, column 9 lines 9-21, column 15 line 39-column 16 line 6, column 8 lines 47-53; column 9 discusses how one time slot is used for the general purpose domain).

30. Referring to claims 7, 27, 36, and 54 the combination of Joy and Ramakrishnan has taught wherein said thread scheduler schedules a non-real-time (NRT) thread to replace a scheduled HRT thread if said HRT is complete (Ramakrishnan column 5 line 54-column 6 line 8, column 9 lines 9-21).

31. Referring to claims 8 and 37 the combination of Joy and Ramakrishnan has taught wherein said thread scheduler schedules the execution of non-real-time (NRT) threads in quanta not allocated to HRT threads (Ramakrishnan column 5 line 54-column 6 line 8, column 9 lines 9-21).

32. Referring to claims 9, 28, 38, and 55 the combination of Joy and Ramakrishnan wherein said thread scheduler regularly schedules NRT threads to be executed (Ramakrishnan column 5 line 54-column 6 line 8, column 9 lines 9-21).

33. Referring to claim 49, Joy has taught allocating available processing time of the processor among at least the first and second threads according to the predetermined fixed execution schedule. See Fig.6, column 2, lines 40-45, and column 3, lines 28-31, and note that the scheduler schedules threads based on cache misses. This scheduling is predetermined in that the system was designed, before runtime, to switch based on cache misses. That is, before the system even begins running a program, it is known that the scheduler will schedule based on cache misses because that is how it was designed. This specific scheduling is also fixed in that if the scheduler is to switch based on cache misses, then if will switch based on cache misses. It will not all of the sudden decide to not switch based on cache misses. Once the hardware is built and programmed to operate as desired, its operation is fixed. It should be noted that the examiner is only referring to the control logic scheduler which performs switching based on cache misses (column 3, lines 28-31).

34. Referring to claim 50, Joy has taught wherein the allocating comprises dividing the available execution time into a plurality of quanta, each quanta corresponding to a number of instruction cycles for execution of a thread (Joy figure 6, column 2 lines 40-45).

35. Referring to claim 51, Joy has taught wherein at least one quanta corresponds to a thread that is scheduled to execute periodically after a fixed number of execution cycles (Joy figure 6, column 2 lines 40-45).

36. Claims 10-12 and 39-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joy in view of Ramakrishnan in further view of Gutgold et al., U.S. Patent Number 6,026,503 (as applied in the previous Office Action and herein referred to as Gutgold).

37. Referring to claims 10 and 39 the combination of Joy and Ramakrishnan has not taught wherein a first storage device for storing program instructions, said processor fetching instructions from the first storage device within a first fetch period;

A second storage device for storing program instructions, said processor fetching instructions from the second storage device within a second fetch period;

Wherein said first fetch period is substantially shorter than said second fetch period. Gutgold has taught wherein a first storage device for storing program instructions, said processor fetching instructions from the first storage device within a first fetch period;

A second storage device for storing program instructions, said processor fetching instructions from the second storage device within a second fetch period;

Wherein said first fetch period is substantially shorter than said second fetch period (Gutgold column 3 lines 1-19; it is well known in the art that flash memory is not as fast in operation as RAM memory, therefore having a slower fetch period than that of the RAM memory fetches).

It would have been obvious to one of ordinary skill in the art at the time of the invention to use two different memories to fetch instructions from. Gutgold has taught storing the debug executable information in the EEPROM, or ROM, and that having a debug mode for a processor is beneficial since it allows the user to test the code they have written just as programmers test code on an emulator (Gutgold column 1 lines 27-45 and column 2 lines 20-26). By being able to

test the code, the user can make sure the code runs efficiently and correctly. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to be able to debug the program code just as an emulator would, allows the program to be tested to make sure that is correct and does not causes errors.

38. Referring to claims 11 and 40 the combination of Joy and Ramakrishnan and Gutgold has taught wherein said first storage device for storing program instructions comprises a static RAM (Gutgold column 3 lines 1-19).

39. Referring to claims 12 and 41 the combination of Joy and Ramakrishnan and Gutgold has taught wherein said second storage device for storing program instructions comprises a flash memory (Gutgold column 3 lines 1-19).

Response to Arguments

40. Applicant's arguments filed on January 24, 2005, have been fully considered but they are not persuasive.

41. Applicant argues the novelty/rejection of claim 1 on pages 13-14 of the remarks, in substance that:

> "Unlike the thread-switching recited in claim 1, the thread switching in Joy is not due to a predetermined fixed schedule; instead Joy's thread-switching is base don some external stimulus or signal, e.g., L1 data cache miss stall signal."

42. These arguments are not found persuasive for the following reasons:

a) The only difference between the currently claimed scheduling and the previously claimed scheduling is that the current scheduling is predetermined. As explained in the rejections above, Joy has taught that the scheduling is both fixed and predetermined. See Fig.6, column 2, lines

40-45, and column 3, lines 28-31, and note that the scheduler schedules threads based on cache misses. This scheduling is predetermined in that the system was designed, before runtime, to switch based on cache misses. That is, before the system even begins running a program, it is known that the scheduler will schedule based on cache misses because that is how it was designed. This specific scheduling is also fixed in that if the scheduler is to switch based on cache misses, then if will switch based on cache misses. It will not all of the sudden decide to not switch based on cache misses. Once the hardware is built and programmed to operate as desired, its operation is fixed. It should be noted that the examiner is only referring to the control logic scheduler which performs switching based on cache misses (column 3, lines 28-31).

43. Applicant argues the novelty/rejection of claim 17 on pages 14-15 of the remarks, in substance that:

> "Joy describes "fast, nanosecond range context switching." However, the context switching described in Joy does not take place between consecutive instruction cycles. However fast it may be, the switching described in Joy produces an overhead of at least one processor cycle. As described in Joy, the thread switch logic supports fast thread switch with a very small delay, for example three cycles or less."

44. These arguments are not found persuasive for the following reasons:

a) As pointed out in the examiner's rejection, column 15, line 52, to column 16, line 5, of Joy discloses logic which allows for "instantaneous switching between threads...". The overhead is disclosed to be one processor cycle, which falls in the ranges of "at least one processor cycle" and "three cycles or less" as argued by applicant. Consequently, if thread A is to execute in cycle X, and a thread switch occurs, then thread B may be executed in cycle X+1. Therefore, it can be seen that switching occurs between consecutive cycles. And, this is further supported by Fig.1B, Fig.2B, and Fig.2C of Joy, because these figures (Fig.2B in particular) show that thread

0 (232) may be switched out and thread 1 (234) switched in with no CPU idle time in between (switch in consecutive cycles). If applicant were correct in saying that switching did not occur in consecutive cycles, then surely there would be CPU idle time between the switching of every thread, which is not the case according to the aforementioned figures.

45. Finally, regarding the 103 rejection arguments that Borkenhagen, Ramakrishnan, and Gutgold do not teach or suggest consecutive-cycle context switching, the examiner asserts that these references do not need to teach this limitation, as Joy has taught it for the reasons explained above.

Conclusion

46. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
April 6, 2005